

COMPUTED TOMOGRAPHY IMAGE RECONSTRUCTION FROM ONLY TWO PROJECTIONS

Ali Mohammad-Djafari

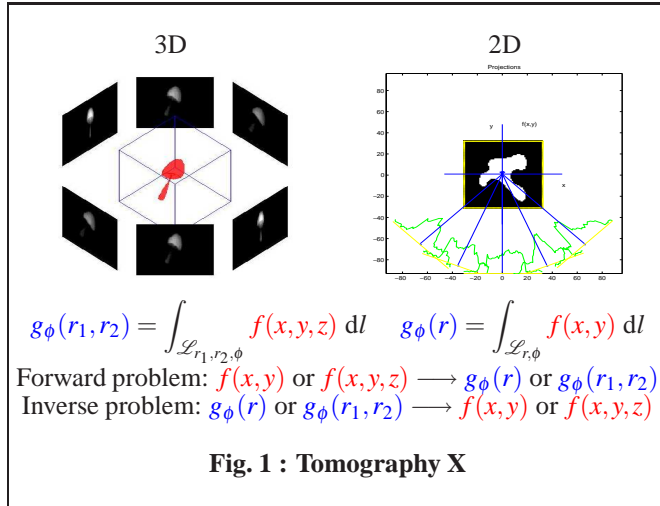
Laboratoire des Signaux et Systèmes,
Unité mixte de recherche 8506 (CNRS-Supélec-UPS)
Supélec, Plateau de Moulon, 3 rue Joliot Curie, 91192 Gif-sur-Yvette, France.

ABSTRACT

This paper concerns the image reconstruction from a few projections in Computed Tomography (CT). The main objective of this paper is to show that the problem is so ill posed that no classical method, such as analytical methods based on inverse Radon transform, nor the algebraic methods such as Least squares (LS) or regularization theory can give satisfactory result. As an example, we consider in detail the case of image reconstruction from two horizontal and vertical projections. We then show how a particular composite Markov modeling and the Bayesian estimation framework can possibly propose satisfactory solutions to the problem. For demonstration and educational purpose a set of Matlab programs are given for a live presentation of the results.

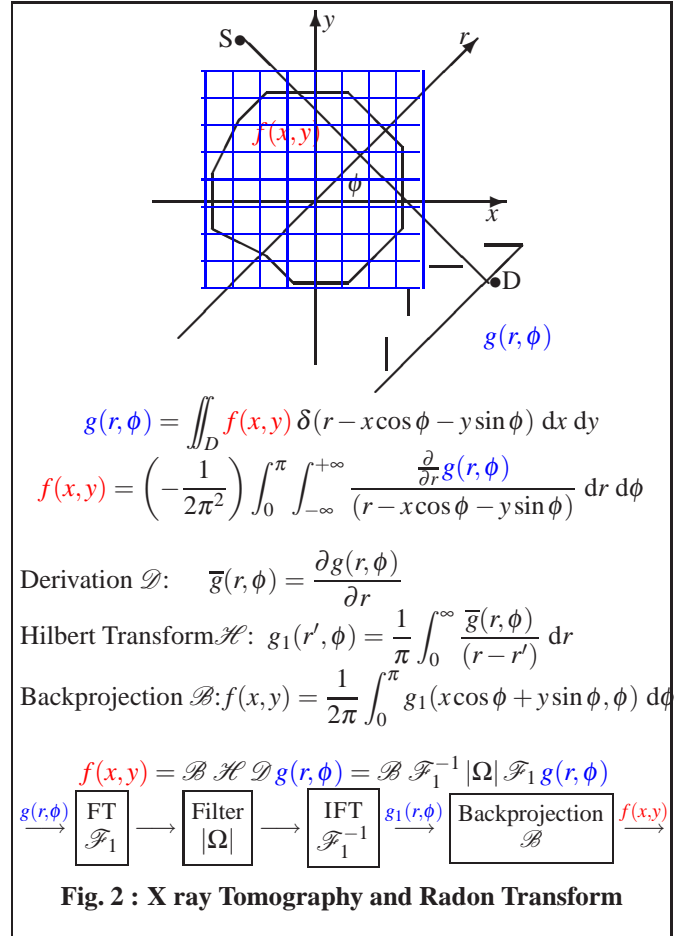
1. INTRODUCTION

The image reconstruction problem in Computed Tomography (CT) is presented in Fig. 1 and Fig. 2 shows the forward modeling of the problem via the Radon Transform (RT) and the basics of the analytical RT inversion based methods such as Filtered backprojection [1, 2, 3, 4, 5, 6, 7, 8].



However, it is so evident that these methods cannot give satisfactory results in cases of very limited number of projections as is the case we consider in this paper. To be able to introduce the necessary prior information needed to overcome the lack of information in the data, we consider the algebraic methods. Fig. 3 shows the discretization step of the forward problem which transforms the linear continuous RT equation to a system of finite linear equations which is $g = Hf$. It is then evident that this system is under-determined and that the problem has an infinite number of

solutions. As a demonstrative example, we consider the case of image reconstruction from only two projections and study the structure of the matrix H in this particular case and show easily that neither the minimum norm least squares (MNLS) nor the generalized inversion and nor the quadratic regularization [9] can give satisfactory result to this problem.

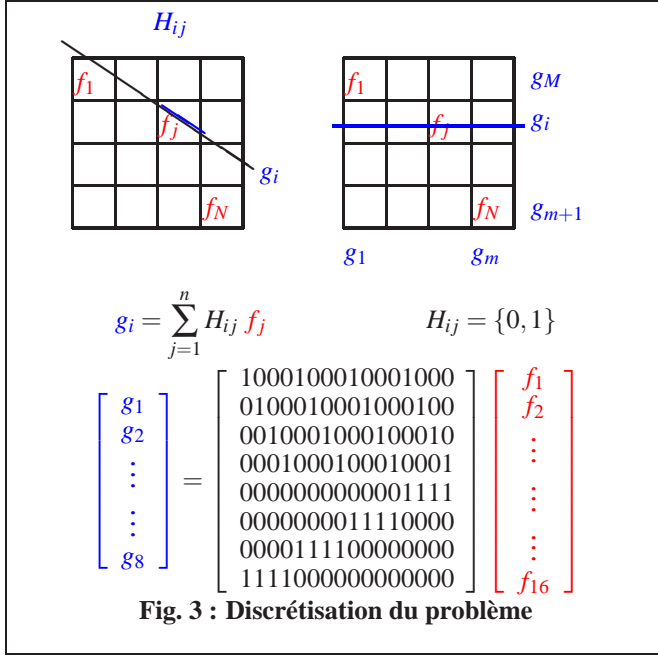


We also show that even applying the positivity constraint is not enough to obtain satisfactory results and that there is a need for more informative prior knowledge. Finally, as the main contribution of this work, we show that the Bayesian inference framework and the composite Markov modeling can possibly be of great help to develop new reconstruction methods which possibly satisfactory results. We consider in particular a composite and hierarchical Intensity-labels Markov modeling with a Gauss-Markov modeling for the intensity

field and a hidden Potts Markov field for the region labels and propose new reconstruction methods which can be applied in many imaging systems, and particularly, in Non Destructive Testing (NDT) imaging applications.

2. DISCRETIZATION OF THE PROBLEM

As we mentioned before, for demonstration purpose, we consider the particular case of image reconstruction from only two projections. Also, for the sake of simplicity, we give details about a very reduced case of a (4×4) pixels image.



If we note by the vector $\mathbf{f} = [f_1, \dots, f_{16}]^t$ the pixels values of the image $f(x, y)$ and by the vector $\mathbf{g} = [g_1, \dots, g_8]^t$ the values of its two projections $g(r, \phi)$ along the horizontal $\phi = 0$ and vertical $\phi = 90$ angles, and assuming $\Delta x = 1, \Delta y = 1, \Delta r = 1$, then we have:

f_1	f_5	f_9	f_{13}	g_8	f_{11}	f_{12}	f_{13}	f_{14}	g_{24}
f_2	f_6	f_{10}	f_{14}	g_7	f_{21}	f_{22}	f_{23}	f_{24}	g_{23}
f_3	f_7	f_{11}	f_{15}	g_6	f_{31}	f_{32}	f_{33}	f_{34}	g_{22}
f_4	f_8	f_{12}	f_{16}	g_5	f_{41}	f_{42}	f_{43}	f_{44}	g_{21}
g_1	g_2	g_3	g_4		g_{11}	g_{12}	g_{13}	g_{14}	

Noting also by $\mathbf{g}_1 = [g_1, \dots, g_4]^t = [g_{11}, \dots, g_{14}]^t$, $\mathbf{g}_2 = [g_5, \dots, g_8]^t = [g_{21}, \dots, g_{24}]^t$ and the matrices \mathbf{A}_1 , \mathbf{A}_2 and \mathbf{A} such that:

$$\mathbf{g}_1 = \mathbf{A}_1 \mathbf{f}, \quad \mathbf{g}_2 = \mathbf{A}_2 \mathbf{f}, \quad \mathbf{g} = \mathbf{A} \mathbf{f} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \mathbf{f}$$

Then, considering the image

$$\mathbf{f} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

the following Matlab code lines:

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix};$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix};$$

$$\mathbf{A} = [\mathbf{A}_1; \mathbf{A}_2];$$

$$\mathbf{f} = [0 \ 0 \ 0 \ 0; 0 \ 1 \ 1 \ 0; 0 \ 1 \ 1 \ 0; 0 \ 0 \ 0 \ 0];$$

$$\mathbf{p} = \mathbf{A} * \mathbf{f}(:);$$

gives :

$$\mathbf{g}' = [02200220]$$

Thus, we have modeled the forward problem. Now, we are going to consider the inverse problem which is given \mathbf{g} find \mathbf{f} .

It is evident that this inverse problem is under-determined, and that it has an infinite number of solutions. Here are four examples:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -.5 & 0 & 0 & .5 \\ 1 & 2 & 0 & -1 \\ -1 & 0 & 2 & 1 \\ 0.5 & 0 & 0 & -.5 \end{bmatrix} \quad \begin{bmatrix} -.5 & 0 & 0 & .5 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ .5 & 0 & 0 & -.5 \end{bmatrix}$$

3. BACKPROJECTION AS THE ADJOINT OPERATOR OF RT AND ITS EQUIVALENT MATRIX TRANSPOSITION

Comparing the continuous RT and its corresponding discretization and the adjoint Backprojection operator given in Fig. 2, and its corresponding discretization, we see easily the equivalence of Backprojection and the transposition of the the matrix \mathbf{A} . Thus, $\hat{\mathbf{f}} = \mathbf{A}' \mathbf{g}$ corresponds to an image obtained by backprojecting the projections and it is easy to see that: $\mathbf{A}' = [\mathbf{A}'_1 \mid \mathbf{A}'_2]$. Thus, computing this solution is easy:

$$\mathbf{f}_h = \mathbf{A}' * \mathbf{p}; \text{reshape}(\mathbf{f}_h, 4, 4)$$

$$\hat{\mathbf{f}} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 2 & 4 & 4 & 2 \\ 2 & 4 & 4 & 2 \\ 0 & 2 & 2 & 0 \end{bmatrix}$$

We may also note that $\hat{\mathbf{f}} = \mathbf{A}' \mathbf{g} = \mathbf{A}'_1 \mathbf{g}_1 + \mathbf{A}'_2 \mathbf{g}_2$ is the addition of two images

$$\hat{\mathbf{f}} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

each being the backprojection of each projection. We may also remark that this solution is not very far from the result

of the convolution of the original image with the following impulse response

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

4. LS, MNLS AND GENERALIZED INVERSION

Let consider the two symetric matrices $A^t A$ and AA^t :

$$AA^t = \left[\begin{array}{c|c} A_1 A_1^t & A_1 A_2^t \\ \hline A_2 A_1^t & A_2 A_2^t \end{array} \right]$$

$$AA^t = \left[\begin{array}{c|c} 4I & 1 \\ \hline 1 & 4I \end{array} \right] = \begin{bmatrix} 4 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 4 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 4 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 4 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 4 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 4 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 4 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 4 \end{bmatrix}$$

$$A^t A = \left[\begin{array}{c|c} A_1^t A_1 & A_2^t A_1 \\ \hline A_1^t A_2 & A_2^t A_2 \end{array} \right]$$

$$A_1^t A_1 = A_2^t A_2 = \left[\begin{array}{c|c} 1+I & I \\ \hline I & 1+I \end{array} \right]$$

$$A_1^t A_2 = A_2^t A_1 = \left[\begin{array}{c|c} I & I \\ \hline I & I \end{array} \right],$$

$$A^t A = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 \end{bmatrix}$$

and compute their singular values:

$$AA^t = A * A^t; \text{svd}(AA^t)$$

$$\text{svd}(AA^t) = [844444440]$$

$$AtA = A^t * A; \text{svd}(AtA);$$

$$\text{svd}(A^t A) = [844444440000000000]$$

We can then remark that both are singular.

We may remind that the least squares (LS) solutions are defined as

$$\hat{f} = \arg \min_f \{ \|g - Af\|^2 \},$$

and if $A^t A$ was invertible, then we had: $\hat{f} = (A^t A)^{-1} A^t g$.

In the same way, the minimum norm solution is defined as:

$$\hat{f} = \arg \min_{Af=g} \{ \|f\|^2 \}$$

and if AA^t was invertible, then we had: $\hat{f} = A^t (AA^t)^{-1} g$.

As we noticed, the two matrices AA^t and $A^t A$ are singular and thus we can not define those solutions. However, if we only consider their diagonal elements, we can define:

$$fh = \text{diag}(1 ./ \text{diag}(AtA)) * A^t * p; \text{reshape}(fh, 4, 4)$$

$$\hat{f} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$fh = A^t * \text{diag}(1 ./ \text{diag}(AA^t)) * p; \text{reshape}(fh, 4, 4)$$

$$\hat{f} = \begin{bmatrix} 0 & .5 & .5 & 0 \\ .5 & 1 & 1 & .5 \\ .5 & 1 & 1 & .5 \\ 0 & .5 & .5 & 0 \end{bmatrix}$$

Also, we can use the technic of truncation of the singular values to define an unique generalized inverse solution

$$\hat{f} = \sum_{k=1}^k \frac{\langle g, u_k \rangle}{\lambda_k} v_k$$

where u_k and v_k are, respectively, the eigenvectors of AA^t and $A^t A$ and λ_k their corresponding eigen values.

$$\begin{aligned} [U, S, V] &= \text{svd}(A); \\ s &= \text{diag}(S); s1 = [1 ./ s(1:7); \text{zeros}(1, 1)]; \\ S1 &= [\text{diag}(s1); \text{zeros}(8, 8)]; \\ fh &= V * S1 * U^t * p; \text{reshape}(fh, 4, 4) \end{aligned}$$

In this example, $K = 7$ and the GI solution can be computed by

$$fh = \text{svdpca}(A, p, .1, 7); \text{reshape}(fh, 4, 4)$$

$$\hat{f} = \begin{bmatrix} -0.2500 & 0.2500 & 0.2500 & -0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ -0.2500 & 0.2500 & 0.2500 & -0.2500 \end{bmatrix}$$

or by the following iterative algorithm:

$$\begin{aligned} \text{for } k &= 1:100; \\ fh &= fh + .1 * A^t * (p - A * fh(:)); \\ \text{end}; \\ \text{reshape}(fh, 4, 4) \end{aligned}$$

$$\hat{f} = \begin{bmatrix} -0.2500 & 0.2500 & 0.2500 & -0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ -0.2500 & 0.2500 & 0.2500 & -0.2500 \end{bmatrix}$$

We may note that the Kernel of $g = Af$, i.e., $\{f | Af = 0\}$ is given by

$$V(I - S^+ S)z = \sum_{k=K+1}^N z_k v_k$$

where z can be any arbitrary image. This can be used to obtain all the possible solutions of $g = Af$ by adding these arbitrary images to the GI solution.

5. REGULARISATION

We may easily note that $A'A + \lambda I$ and $AA' + \lambda I$ are no more singular if $\lambda > 0$. Thus, we may compute:

```
lambda=.01;
fh=inv(AtA+lambda*eye(size(AtA)))*(A'*p);
reshape(fh,4,4)
```

$$\hat{f} = \begin{bmatrix} -0.2491 & 0.2497 & 0.2497 & -0.2491 \\ 0.2497 & 0.7484 & 0.7484 & 0.2497 \\ 0.2497 & 0.7484 & 0.7484 & 0.2497 \\ -0.2491 & 0.2497 & 0.2497 & -0.2491 \end{bmatrix}$$

or still

```
lambda=.01;
fh=A'*inv(AA+lambda*eye(size(AA)))*p;
reshape(fh,4,4)
```

which give the same solution.

6. POSITIVITY CONSTRAINT

We may remark that the problem is so ill-posed that, imposing to the solutions to be of the minimum norme does not reduce enough the space of the possible solutions. The positivity constraint has been frequently used in many image reconstruction applications. A very simple technique to impose the positivity constraint in iterative algorithms is just to impose it at each iteration:

```
for k=1:100
fh=fh+.1*A'*(p-A*fh(:));
fh=fh.*(fh>0);
end
reshape(fh,4,4);
```

$$fh = \begin{bmatrix} 0 & 0.0000 & 0.0000 & 0 \\ 0.0000 & 1.0000 & 1.0000 & 0.0000 \\ 0.0000 & 1.0000 & 1.0000 & 0.0000 \\ 0 & 0.0000 & 0.0000 & 0 \end{bmatrix}$$

Of course, this technique is only one of the possible methods to use the prior information of the positivity. However, we see that it can be very useful at least in this low scale case. But, as we will see later, in a real larger image reconstruction problem, it is not enough.

7. REAL SIZE IMAGES IMPLEMENTATION OF THE ALGORITHMES

Let consider a (256×256) pixel image. Then, it is no more question of really constructing the matrix A , because its dimensions are $(256^2 \times 256)$. Indeed, we do not really need its construction, we only need the results of the forward computation Af and the backprojection $A'g$. The following programs shows how to compute these quantities without actually constructing the matrix A .

```
function p=direct(f);
p1=sum(f);
p2=sum(f');
p=[p1(:);p2(:)];
return
```

```
function f=transp(p);
l=length(p);p1=p(1:l/2);p2=p(l/2+1:l);
f=ones(l/2,1)*p1'+p2*ones(1,l/2);
return
```

These programs can easily be used to obtain the following results/

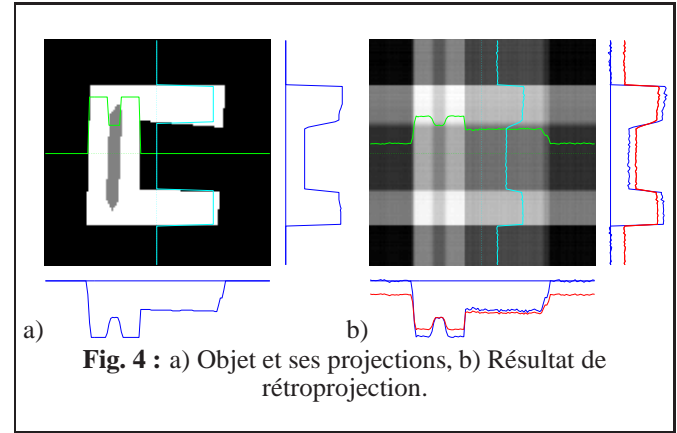


Fig. 4 : a) Objet et ses projections, b) Résultat de rétroprojection.

We can also use them to impose any constraints such as positivity in the iterative methods:

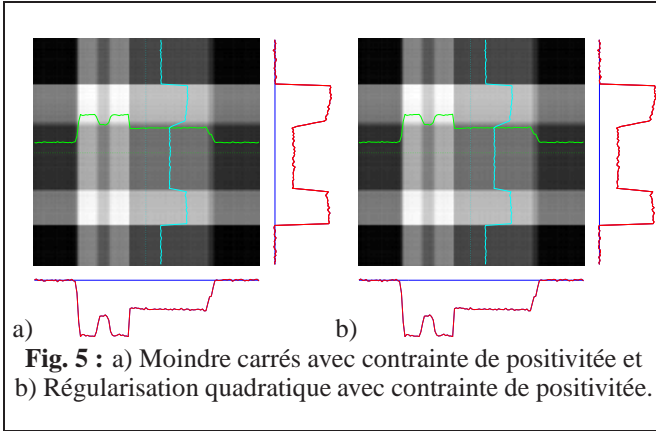
Moindre carré avec contrainte de positivité :

```
alpha=.1;
for k=1:100
g=trans(p-direct(fh));
fh=fh+alpha*g;
fh=fh.*(fh>0);
end
```

Régularisation quadratique avec contrainte de positivité :

```
alpha=.1;d=[-1 0 -1;0 4 0;-1 0 -1];
for k=1:100
g0=trans(p-direct(fh));
g=g0-lambda*conv2(fh,d,'same');
fh=fh+alpha*g;
fh=fh.*(fh>0);
end
```

Remarquons que les algorithmes présentés plus haut sont assez rudimentaires (gradient à pas constant et à nombre d'itérations fini). Il est évident que l'on peut faire mieux. À titre d'exemple, nous avons développé un logiciel d'optimisation (gpave) un peu plus élaboré qui met en œuvre d'autres algorithmes d'optimisation, comme par exemple, gradient à pas adaptative, gradient conjugué et d'autres encore.



Les lignes de codes Matlab qui suivent montrent l'usage de ce logiciel. Il faut tout d'abord écrire deux routines qui calculent le critère `crit` qui calcule

$$J = \|g - Af\|^2 + \lambda \|Df\|^2$$

et son gradient `dcrit`

$$\nabla J = -2A^t(g - Af) + 2\lambda D^t Df$$

où Df correspond à l'application d'une opération de convolution de l'image $f(i, j)$ avec une réponse impulsionnelle

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \text{ ce qui correspond à}$$

$$\sum_i \sum_j (|f(i, j) - f(i-1, j)|^2 + |f(i, j) - f(i, j-1)|^2).$$

Notez aussi que $D^t Df$ correspond à l'application d'une opération de convolution de l'image $f(i, j)$ avec une réponse impulsionnelle

$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & -4 & 0 \\ -1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} * \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

où $*$ signifie une convolution.

```
function J=crit(fh,p,lambda)
dp=p-direct(fh);
J0=sum(dp(:).^2);
d=[-1 1;-1 1];
df=conv2(fh,d,'same');
J1=sum(df(:).^2);
J=J0+lambda*J1;
return

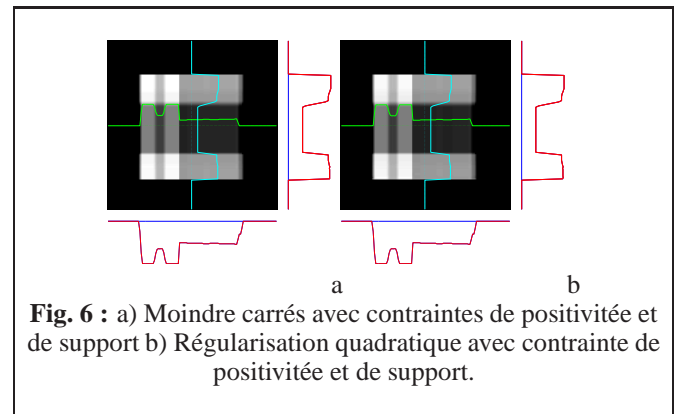
function dJ=dcrit(fh,p,lambda)
dp=p-direct(fh);
dJ0=-2*transp(dp);
d=[-1 0 -1;0 4 0;-1 0 -1];
dJ1=conv2(fh,d,'same');
dJ=dJ0+lambda*dJ1;
return
```

Avec ces deux routines, le programme de la reconstruction devient très simple:

```
f0=transp(p);
options = options;
lambda=1;
fh=gpav('crit',f0,options,'dcrit',p,lambda)
```

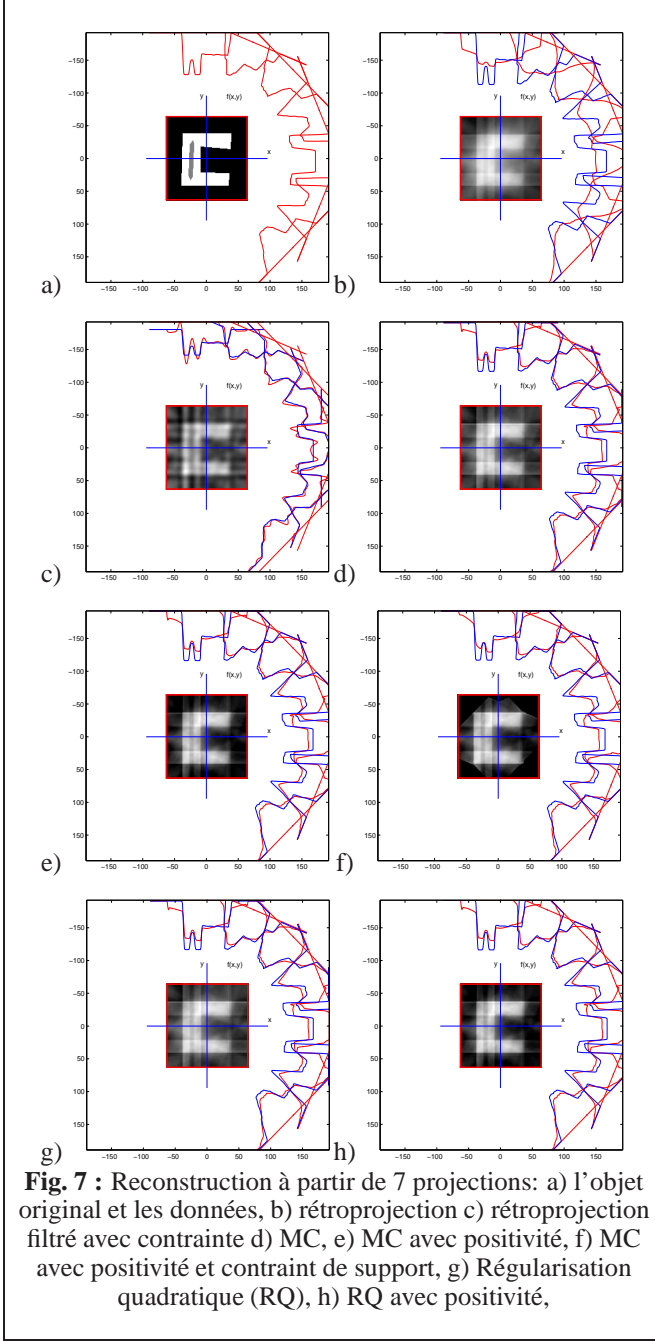
`reshape(fh,4,4)`

Avec ce programme d'optimisation il est alors facile de modifier les routines `crit` et `dcrit` pour changer le critère de la régularisation. Les figures suivantes montrent un certain nombre des résultats.



On remarque que le problème est très mal-conditionnée au sens que la manque d'information dans les données est trop important. Il faut pouvoir obtenir d'autres données, *i.e.*, des projections suivant d'autre angles. Pour cela il faut réécrire les routines `directe` et `transp` et les routines correspondantes `crit` et `dcrit` afin de pouvoir implémenter le cas général du calcul des projections suivant n'importe quel angle et la rétroprojection associée.

Les figures suivantes montrent des exemples de reconstructions pour le cas où on a 7 projections.



8. MODÉLISATION PAR CHAMPS DE MARKOV COMPOSITES

Évidament, plus on a des données bien réparties, mieux sera les résultats. Mais, lorsque l'obtention d'autres projections est impossible, il faudra récompenser la manque d'information par des modélisations plus précises. En particulier, dans le domaine du contrôle non destructif (CND), une information *a priori* importante est que l'objet est composé d'un nombre fini de matériaux. Ceci signifie que l'image que nous cherchons à reconstruire est composée d'un nombre fini de régions homogènes. C'est exactement la modélisation de cette information *a priori* qui est l'originalité des travaux que nous menons dans notre laboratoire.

L'outil est la modélisation probabiliste par champs de

Markov et l'estimation bayésienne. Un grand nombre de travaux ont été fait sur ce sujet (voir par exemple [10, 11, 12]). Ici, nous mentionnons seulement deux modélisations : Modélisation de l'image par un champs composite (intensités-contours) ou (intensités-régions). Dans la première, on introduit une variable cachée binaire $q(\mathbf{r})$ qui représente les contours et dans la deuxième on introduit une variable cachée discrète $z(\mathbf{r})$ qui peut prendre des valeurs discrètes $k = 1, \dots, K$, représentant les labels attribués aux pixels $f(\mathbf{r})$ de l'image ayant les mêmes propriétés (par exemple se trouvant dans une même région homogène).

8.1 Modèle Intensités-Contours

Dans cette modélisation, l'idée de base est de modéliser le fait qu'une image est en faite une fonction $f(\mathbf{r})$ qui est continue par morceaux (piecewise continuous) ou par régions. Il y a donc des discontinuités (contours). On peut alors modéliser ces contours par une image binaire $q(\mathbf{r})$. Le point essentiel est alors de décrire à l'aide d'une loi de probabilité conditionnelle $p(\mathbf{f}|\mathbf{q})$, le lien qu'il y entre des variables intensités \mathbf{f} et des variables contours \mathbf{q} qui peut être résumé par :

Cas 1D:

$$p(f_j|q_j, f_i, i \neq j) = \mathcal{N}(\beta(1 - q_j)f_{j-1}, \sigma_f^2)$$

Cas 2D:

$$p(f(\mathbf{r})|q(\mathbf{r}), f(\mathbf{s})) = \mathcal{N}\left(\beta(1 - q(\mathbf{r})) \sum_{\mathbf{s} \in \mathcal{V}(\mathbf{r})} f(\mathbf{s}), \sigma_f^2\right)$$

Ensuite, en choisissant une loi *a priori* appropriée pour $p(\mathbf{q})$ et en utilisant des lois $p(\mathbf{g}|\mathbf{f})$ et $p(\mathbf{f}|\mathbf{q})$, on obtien la loi *a posteriori* $p(\mathbf{f}, \mathbf{q}|\mathbf{g})$ qui peut être utilisée pour inferer conjointement sur \mathbf{f} et sur \mathbf{q} . A titre d'indication, considérons l'estimation au sens du MAP :

$$(\hat{\mathbf{f}}, \hat{\mathbf{q}}) = \arg \max_{\mathbf{f}, \mathbf{q}} \{p(\mathbf{f}, \mathbf{q}|\mathbf{g})\}$$

qui peut être obtenu par un algorithme itératif du type :

$$\begin{aligned} \hat{\mathbf{f}} &= \arg \max_{\mathbf{f}} \{p(\mathbf{f}|\mathbf{g}, \mathbf{q})\} = \arg \min_{\mathbf{f}} \{J(\mathbf{f})\} \\ \hat{\mathbf{q}} &= \arg \max_{\mathbf{q}} \{p(\mathbf{q}|\mathbf{g})\} \end{aligned}$$

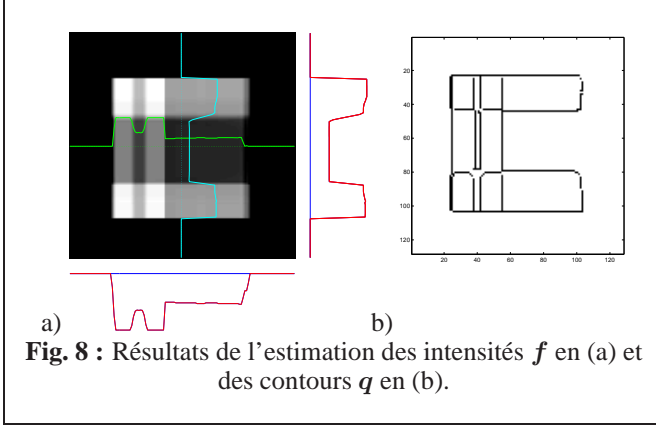
avec

$$J(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \sum_{\mathbf{r}} (1 - q(\mathbf{r})) \left(f(\mathbf{r}) - \beta \sum_{\mathbf{s} \in \mathcal{V}(\mathbf{r})} f(\mathbf{s}) \right)^2$$

L'étape difficile est l'obtention de l'expression de $p(\mathbf{q}|\mathbf{g})$ et surtout son optimisation, qui idéalement ne peut se faire qu'à l'aide d'une recherche combinatoire. Il existe un très grand nombre de travaux portant sur différentes approximations qui permettent d'effectuer cette optimisation d'une manière approchée mais réaliste en coût de calcul pour des applications réelles.

Pour plus de détail sur cette méthode se référer à [10, 11, 12].

Ici, nous montrons un résultat typique que l'on peut obtenir avec de telles méthode. Comme nous pouvons constater, cette modélisation *a priori* n'est pas encore suffisamment forte pour obtenir un résultat satisfaisant pour ce Inverse problem très difficile.



8.2 Modèle Intensités-Régions

La modélisation précédente, bien que déjà plus spécifique, n'apportait pas d'information sur des valeurs qui peuvent prendre des pixels de l'image. Dans certaines applications, par exemple en contrôl non destructif (CND), nous savons *a priori* que l'objet est composé d'un nombre fini de matériaux (air, métal, composite). La modélisation qui suit permet de prendre en compte ce type d'information *a priori*.

Plus spécifiquement, nous proposons de modéliser l'image par un champs composite (intensités-labels), où les labels $z(\mathbf{r})$ représentent la nature de matériaux à la position du pixel \mathbf{r} et l'homogénéité des intensités $\mathbf{f}_k = \{f(\mathbf{r}), \mathbf{r} \in \mathcal{R}_k\}$ dans une région données $\mathcal{R}_k = \{\mathbf{r} : z(\mathbf{r}) = k\}$ est modélisé par un champs de Gauss-Markov:

$$p(\mathbf{f}_k) = \mathcal{N}(m_k \mathbf{1}, \Sigma_k)$$

ce qui peut être interprété aussi par des relations suivantes:

$$\begin{aligned} p(f(\mathbf{r})|z(\mathbf{r}) = k) &= \mathcal{N}(m_k, v_k) \\ p(f(\mathbf{r})) &= \sum_{k=1}^K p(z(\mathbf{r}) = k) \mathcal{N}(m_k, \sigma_k^2) \\ \Sigma_k &= \text{diag}[v_1, \dots, v_K] \end{aligned}$$

ce qui montre que la distribution marginale de chaque pixel de l'images est modélisée par un mélange de gaussiennes.

Supposant ensuite qu'*a priori* les pixels qui se trouvent dans deux régions différentes soient indépendantes, on peut écrire :

$$p(\mathbf{f}|\mathbf{z}) = \prod_{k=1}^K \mathcal{N}(m_k \mathbf{1}, \Sigma_k)$$

La particularité de la méthode que nous proposons est une modélisation spécifique pour des labels des régions (Champs de Potts) qui permet de décrire pour $p(\mathbf{z})$

$$p(\mathbf{z}) \propto \exp \left[\alpha \sum_{\mathbf{r} \in \mathcal{R}} \sum_{\mathbf{s} \in \mathcal{V}(\mathbf{r})} \delta(z(\mathbf{r}) - z(\mathbf{s})) \right]$$

Notant aussi par $\theta = \{\sigma_\varepsilon^2, (m_k, \sigma_k^2), k = 1, \dots, K\}$ que l'on appelle le vecteur des hyperparamètres, on aura à exprimer la loi *a posteriori* jointe $p(\mathbf{f}, \mathbf{z}, \theta | \mathbf{g})$, qui peut ensuite être utilisée pour estimer ces inconnues. Différents choix sont alors possibles. Ici, nous mentionnons ceux que nous avons implémentés et utilisés:

- MAP (Algorithm 1):

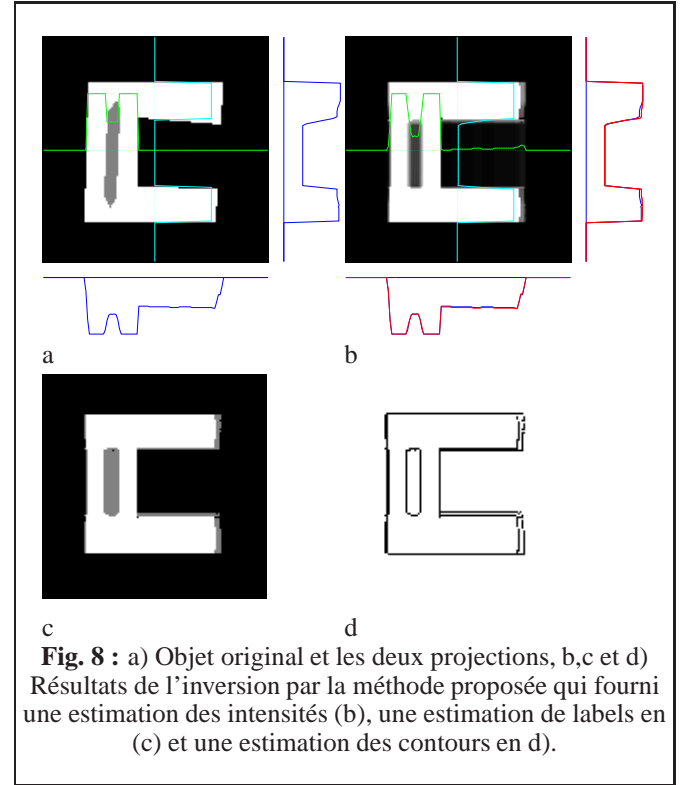
$$\begin{cases} \hat{\mathbf{f}} &= \arg \max_{\mathbf{f}} \{p(\mathbf{f}|\mathbf{z}, \theta, \mathbf{g})\} \\ \hat{\theta} &= \arg \max_{\theta} \{p(\theta|\mathbf{f}, \mathbf{z}, \mathbf{g})\} \\ \hat{\mathbf{z}} &= \arg \max_{\mathbf{z}} \{p(\mathbf{z}|\mathbf{f}, \theta, \mathbf{g})\} \end{cases}$$

- MAP-Gibbs (Algorithm 2):

$$\begin{cases} \hat{\mathbf{f}} &= \arg \max_{\mathbf{f}} \{p(\mathbf{f}|\mathbf{z}, \theta, \mathbf{g})\} \\ \text{échant. } \hat{\theta} &\text{avec } p(\theta|\mathbf{f}, \mathbf{z}, \mathbf{g}) \\ \text{échant. } \hat{\mathbf{z}} &\text{avec } p(\mathbf{z}|\mathbf{f}, \theta, \mathbf{g}) \end{cases}$$

Pour plus de détails sur les expressions de ces lois et la mise en oeuvre de la méthode dans un cadre plus général se référer à [13, 14, 15, 16].

Principal avantage d'une telle modélisation et d'un tel méthode est que l'on obtient non seulement une estimation de \mathbf{f} mais aussi de \mathbf{z} qui représente une segmentation de l'image, et aussi par un simple algorithme de détection de contours sur \mathbf{z} on obtiendra aussi une image des contours \mathbf{q} . La figure qui suit montre un résultat typique.



9. CONCLUSION

Dans ce travail, à but pédagogique, au travers d'un Inverse problem de la reconstruction d'image en Tomography X lorsque le nombre de projections sont très limité, nous avons analysé les difficultés inhérentes des problèmes inverses. Le principal objectif était de montrer que les différentes méthodes classiques naïves, mais très utilisées, ne donnent pas de solutions satisfaisantes et qu'il y a un besoin de proposer des méthodes d'inversion plus sophistiquées qui permettent d'introduire de l'information *a priori* nécessaire pour compenser la manque d'information dans les données.

Un grand nombre de modélisations ont été proposées (voir par exemple [17, 18]). Mais, ici, nous nous sommes contentés des méthodes qui modélisent l'image au niveau des pixels.

Le cas particulier de la reconstruction à partir de deux projections est détaillé et une méthode basée sur la modélisation de l'image par un champs de Markov composé (intensité-labels) et l'estimation bayésienne est présentée qui

permet, au moins, d'obtenir une solution satisfaisante au problème. Les routines Matlab correspondant est disponible sur <http://djafari.free.fr/TomoX>.

REFERENCES

- [1] G. T. Herman and S. W. Rowland, "Three methods for reconstructing objects from X-rays: A comparative study," *Computer Graphics and Image Processing*, vol. 2, pp. 151–178, 1973.
- [2] K. M. Hanson, "On the optimality of the filtered backprojection algorithm," *J. Comput. Assist. Tomogr.*, vol. 4, pp. 361–363, 1980.
- [3] G. T. Herman, *Image reconstruction from projections. The fundamentals of computerized tomography*. New York, NY: Academic Press, 1980.
- [4] P. Grangeat, "Mathematical framework of cone beam 3D reconstruction via the first derivative of the Radon transform," in *Mathematical Methods in Tomography* (G. T. Herman, A. K. Louis, and F. Natterer, eds.), vol. 1497, (New York), pp. 66–97, Springer Verlag, 1991.
- [5] F. Natterer, "Algorithms in tomography," in *The State of the Art in Numerical Analysis*, Clarendon Press, duff, i.s. and watson, g.a. ed., 1997.
- [6] A. Mohammad-Djafari and C. Soussen, "Compact object reconstruction," in *Discrete Tomography: Foundations, Algorithms and Applications* (G. T. Herman and A. Kuba, eds.), ch. 14, pp. 317–342, Boston, MA: Birkhauser, 1999.
- [7] F. Natterer, "Numerical methods in tomography," in *Acta Numerica*, vol. 8, Cambridge University Press, July 1999.
- [8] Ali Mohammad-Djafari and Jean-Marc Dintin, "Reconstruction tomographique à partir d'un nombre faible de projections," in *Approche bayésienne pour les problèmes inverses* (Japan, ed.), (Paris), pp. 297–320, Traité IC2, Série traitement du signal et de l'image, Hermès, 2001.
- [9] G. Demoment, J. Idier, J.-F. Giovannelli, and A. Mohammad-Djafari, "Problèmes inverses en traitement du signal et de l'image," vol. TE 5 235 of *Traité Télécoms*, pp. 1–25, Paris, France: Techniques de l'Ingénieur, 2001.
- [10] M. Nikolova, A. Mohammad-Djafari, and J. Idier, "Inversion of large-support ill-conditioned linear operators using a Markov model with a line process," in *Proc. IEEE ICASSP*, vol. V, (Adelaide, Australia), pp. 357–360, 1994.
- [11] S. S. Saquib, C. A. Bouman, and K. D. Sauer, "ML parameter estimation for Markov random fields with applications to Bayesian tomography," *IEEE Trans. Image Processing*, vol. 7, pp. 1029–1044, July 1998.
- [12] A. Mohammad-Djafari, "Hierarchical markov modeling for fusion of x ray radiographic data and anatomical data in computed tomography," in *Int. Symposium on Biomedical Imaging (ISBI 2002)*, 7-10 Jul., Washington DC, USA, July 2002.
- [13] Hichem Snoussi AND Ali Mohammad-Djafari, "Bayesian unsupervised learning for source separation with mixture of gaussians prior," *Journal of VLSI Signal Processing Systems*, vol. 37, pp. 263–279, June/July 2004.
- [14] Hichem Snoussi AND Ali Mohammad-Djafari, "Fast joint separation and segmentation of mixed images," *Journal of Electronic Imaging*, vol. 13, pp. 349–361, April 2004.
- [15] O. Féron and A. Mohammad-Djafari, "Image fusion and joint segmentation using an MCMC algorithm," *Journal of Electronic Imaging*, vol. 14, p. paper no. 023014, Apr 2005.
- [16] F. Humblot and A. Mohammad-Djafari, "Super-Resolution using Hidden Markov Model and Bayesian Detection Estimation Framework," *EURASIP Journal on Applied Signal Processing*, vol. Special number on Super-Resolution Imaging: Analysis, Algorithms, and Applications, pp. ID 36971, 16 pages, 2006.
- [17] A. Kuba and G. T. Herman, "Discrete tomography: A historical overview," in *Discrete Tomography: Foundations, Algorithms and Applications* (G. T. Herman and A. Kuba, eds.), ch. 1, pp. 3–34, Boston, MA: Birkhauser, 1999.
- [18] Charles Soussen and Ali Mohammad-Djafari, "Polygonal and polyhedral contour reconstruction in computed tomography," *IEEE Trans. on Image Processing*, vol. 13, pp. 1507–1523, Nov 2004.

RECONSTRUCTION D'IMAGE À PARTIR DE DEUX PROJECTIONS EN TOMOGRAPHIE X

Ali Mohammad-Djafari

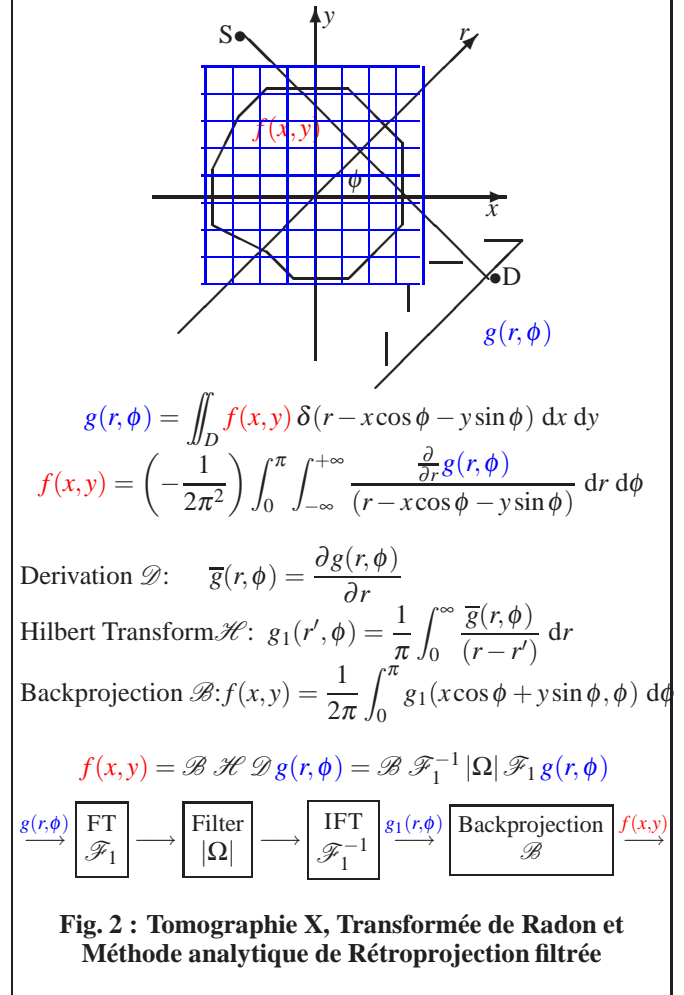
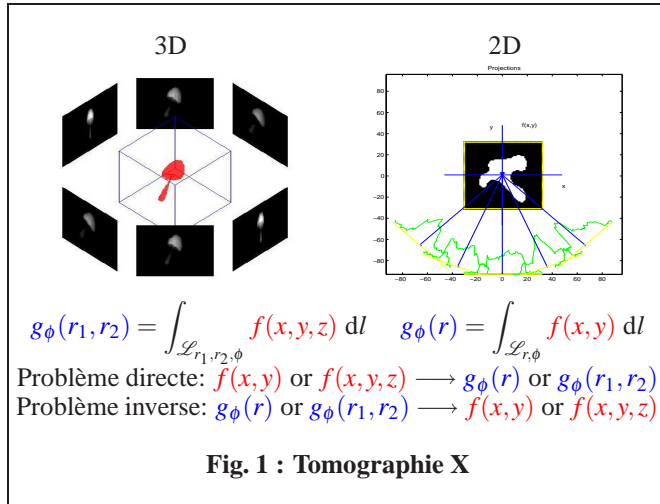
Laboratoire des Signaux et Systèmes,
Unité mixte de recherche 8506 (CNRS-Supélec-UPS)
Supélec, Plateau de Moulon, 3 rue Joliot Curie, 91192 Gif-sur-Yvette, France.

ABSTRACT

Ce travail, à but pédagogique, présente le problème inverse de la reconstruction d'image en tomographie X lorsque le nombre des projections est très limité. L'objectif est de montrer que les différentes méthodes classiques naïves, mais très utilisées pour sa résolution, ne donnent pas de solutions satisfaisantes. Il est alors nécessaire de proposer des méthodes d'inversion plus sophistiquées qui permettent d'introduire de l'information *a priori* nécessaire pour l'obtention d'une solution acceptable. Le cas particulier de la reconstruction à partir de deux projections est détaillé, les résultats que l'on peut obtenir avec les différentes méthodes algébriques sont présentés et la nécessité des méthodes probabilistes est démontrée. Finalement, une méthode basée sur la modélisation de l'image par un champs de Markov composé (intensité-labels) et l'estimation bayésienne est présentée qui permet, au moins, d'obtenir une solution satisfaisante au problème. Dans ce travail, l'accent est mis sur l'aspect pédagogique. En effet, ce problème est traité dans le cadre d'un enseignement au niveau du DEA et les outils (programmes Matlab) nécessaires pour une démonstration seront disponibles lors de l'exposé.

1. INTRODUCTION

Le problème de la reconstruction d'image en tomographie X est présenté dans Fig. 1 et Fig. 2 montre la modélisation du problème via la Transformée de Radon et le schéma de la méthode classique de rétroprojection filtrée [1, 2, 3, 4, 5, 6, 7].

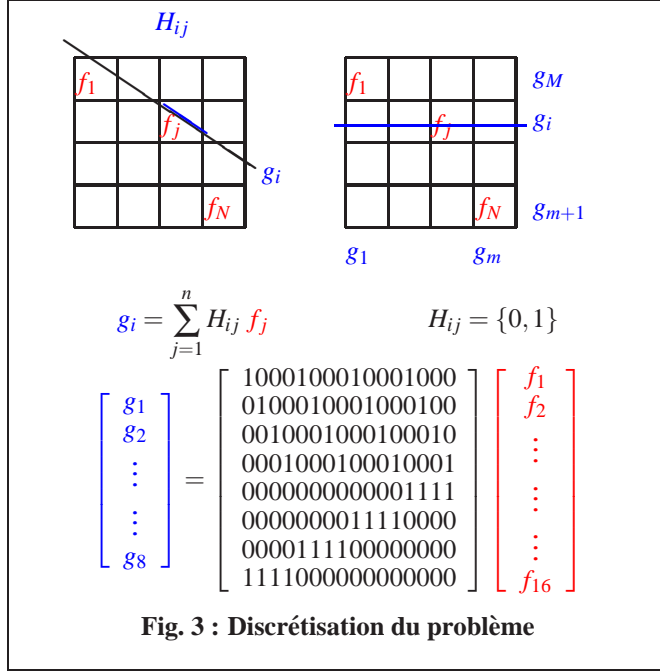


Il est évident que ces méthodes analytiques (Rétroprojection ou Rétroprojection filtrée) ne donnent des résultats satisfaisants que lorsqu'il y a un grand nombre de projections. C'est pourquoi, dans ce travail, nous considérons les méthodes algébriques qui permettent plus de souplesse pour le développement des méthodes plus sophistiquées. Fig. 3 montre l'étape de la discrétisation du problème qui le transforme à celui de la résolution d'un système d'équations linéaires $\mathbf{g} = \mathbf{H}\mathbf{f}$. Nous montrons ensuite que le problème est sous-déterminé et qu'il y a une infinité de solutions possibles. Plus intéressant encore qu'aucune méthode algébrique du type Moindres carrés (MC), Inversion généralisée, ou même la régularisation quadratique [8, 9], ne fournit une solution satisfaisante. Aussi, l'application des contraintes de positivité, bien

qu'améliore les résultats de ces méthodes, ne suffit pas, d'où la nécessité de proposer des modélisations *a priori* plus complexes.

2. DISCRETIZATION DU PROBLÈME

Dans ce travail, afin de pédagogie, nous considérons un problème de reconstruction d'image avec de dimensions réduites à partir de deux projections horizontale et verticale, ce qui permet de mieux appréhender les difficultés du problème.



Soient $\mathbf{f} = [f_1, \dots, f_{16}]^t$ les valeurs des pixels de l'image $f(x, y)$ de dimensions (4×4) et $\mathbf{g} = [g_1, \dots, g_8]^t$ les valeurs de ses projections $g(r, \phi)$ suivant les angles $\phi = 0$ et $\phi = 90$ degrés. Supposons $\Delta x = 1, \Delta y = 1$ et considérons les deux représentations suivantes:

f_1	f_5	f_9	f_{13}	g_8	f_{11}	f_{12}	f_{13}	f_{14}	g_{24}
f_2	f_6	f_{10}	f_{14}	g_7	f_{21}	f_{22}	f_{23}	f_{24}	g_{23}
f_3	f_7	f_{11}	f_{15}	g_6	f_{31}	f_{32}	f_{33}	f_{34}	g_{22}
f_4	f_8	f_{12}	f_{16}	g_5	f_{41}	f_{42}	f_{43}	f_{44}	g_{21}
g_1	g_2	g_3	g_4		g_{11}	g_{12}	g_{13}	g_{14}	

Notons aussi

$$\mathbf{g}_1 = [g_1, \dots, g_4]^t = [g_{11}, \dots, g_{14}]^t,$$

$$\mathbf{g}_2 = [g_5, \dots, g_8]^t = [g_{21}, \dots, g_{24}]^t$$

et formons les matrices \mathbf{A}_1 , \mathbf{A}_2 et \mathbf{A} telle qu'on puisse écrire

$$\mathbf{g}_1 = \mathbf{A}_1 \mathbf{f}, \quad \mathbf{g}_2 = \mathbf{A}_2 \mathbf{f}, \quad \mathbf{g} = \mathbf{A} \mathbf{f} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \mathbf{f}$$

Considérons l'image

$$\mathbf{f} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

et calculons sa projection en utilisant les lignes Matlab suivantes :

```
A1=[ 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1];
```

```
A2=[ 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1;
0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0;
0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0;
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0];
```

```
A=[A1;A2];
```

```
f=[0 0 0 0;0 1 1 0;0 1 1 0;0 0 0 0];
```

```
p=A*f(:);
```

ce qui donne :

$$\mathbf{g}^t = [02200220]$$

Ainsi, la résolution du problème direct ne pose aucune difficulté. Mais, considérons maintenant le problème inverse: étant donnée \mathbf{g} trouver \mathbf{f} .

3. INDÉTERMINATION DU PROBLÈME

Il est évident que ce problème a une infinité de solutions possibles. En voici quatre:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -.5 & 0 & 0 & .5 \\ 1 & 2 & 0 & -1 \\ -1 & 0 & 2 & 1 \\ 0.5 & 0 & 0 & -.5 \end{bmatrix} \quad \begin{bmatrix} -.5 & 0 & 0 & .5 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ .5 & 0 & 0 & -.5 \end{bmatrix}$$

4. ÉQUIVALENCE ALGÈBRE DE RÉTROPROJECTION

En comparant les relation continue et discrète et les opérateurs direct et adjoint, on trouve que l'opération de rétroprojection en continue correspond à l'opérateur de transposition de matrice en discret. Ainsi, la solution $\hat{\mathbf{f}} = \mathbf{A}^t \mathbf{g}$ correspond à la solution au sens de la la rétro-projection. Exprimons alors la matrice $\mathbf{A}^t = [\mathbf{A}_1^t \mid \mathbf{A}_2^t]$ et calculons cette solution :

```
fh=A'*p;reshape(fh,4,4)
```

$$\hat{\mathbf{f}} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 2 & 4 & 4 & 2 \\ 2 & 4 & 4 & 2 \\ 0 & 2 & 2 & 0 \end{bmatrix}$$

Notons que $\hat{\mathbf{f}} = \mathbf{A}^t \mathbf{g} = \mathbf{A}_1^t \mathbf{g}_1 + \mathbf{A}_2^t \mathbf{g}_2$ est l'addition de deux images

$$\hat{\mathbf{f}} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

chacune étant la rétro-projection d'une des deux projections.

Remarquons aussi que cette image, à une constant près est très proche du résultat de la convolution de l'image d'origine avec la réponse impulsionnelle

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

5. INVERSION GÉNÉRALISÉE

En vue de la définition d'une solution au sens d'inversion généralisée, calculons les matrices $A^t A$ et AA^t :

$$AA^t = \left[\begin{array}{c|c} A_1 A_1^t & A_1 A_2^t \\ \hline A_2 A_1^t & A_2 A_2^t \end{array} \right]$$

$$AA^t = \left[\begin{array}{c|c} 4I & 1 \\ \hline 1 & 4I \end{array} \right] = \begin{bmatrix} 4 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 4 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 4 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 4 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 4 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 4 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 4 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 4 \end{bmatrix}$$

$$A^t A = \left[\begin{array}{c|c} A_1^t A_1 & A_2^t A_1 \\ \hline A_1^t A_2 & A_2^t A_2 \end{array} \right]$$

$$A_1^t A_1 = A_2^t A_2 = \left[\begin{array}{c|c} 1+I & I \\ \hline I & 1+I \end{array} \right]$$

$$A_1^t A_2 = A_2^t A_1 = \left[\begin{array}{c|c} I & I \\ \hline I & I \end{array} \right],$$

$$A^t A = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 1 \end{bmatrix}$$

Calculons les valeurs singulières des matrices AA^t et $A^t A$:

$$AA^t = A * A^t; \text{ svd}(AA^t)$$

$$\text{svd}(AA^t) = [844444440]$$

$$A^t A = A^t * A; \text{ svd}(A^t A);$$

$$\text{svd}(A^t A) = [844444440000000000]$$

6. MOINDRES CARRÉS DE NORME MINIMAL, INVERSION GÉNÉRALISÉE ET DTVS

Notons que ces deux matrices sont singulières. Rappelons qu'une solution au sens des moindres carrés s'écrit:

$$\hat{f} = \arg \min_f \{ \|g - Af\|^2 \},$$

et si la matrice $A^t A$ est inversible on obtient : $\hat{f} = (A^t A)^{-1} A^t g$.

De même, une solution de norme minimale est

$$\hat{f} = \arg \min_{Af=g} \{ \|f\|^2 \}$$

et si la matrice AA^t est inversible on obtient : $\hat{f} = A^t (AA^t)^{-1} g$.

Nous ne pouvons alors calculer aucune de ces deux solutions car aucune des deux matrices AA^t et $A^t A$ est inversible. Notons cependant que si on ne garde que les éléments diagonaux de ces deux matrices, on obtient des résultats suivants :

$$fh = \text{diag}(1 ./ \text{diag}(A^t A)) * A^t * p; \text{reshape}(fh, 4, 4)$$

$$\hat{f} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$fh = A^t * \text{diag}(1 ./ \text{diag}(AA^t)) * p; \text{reshape}(fh, 4, 4)$$

$$\hat{f} = \begin{bmatrix} 0 & .5 & .5 & 0 \\ .5 & 1 & 1 & .5 \\ .5 & 1 & 1 & .5 \\ 0 & .5 & .5 & 0 \end{bmatrix}$$

Il est cependant possible de calculer la solution inverse généralisée qui est la solution de norme minimale de $A^t A f = A^t g$ en utilisant la décomposition tronquée des valeurs singulières (DTVS):

$$\hat{f} = \sum_{k=1}^k \frac{\langle g, u_k \rangle}{\lambda_k} v_k$$

où u_k et v_k sont, respectivement, des vecteurs propres de AA^t et de $A^t A$ et λ_k sont des valeurs singulières associées.

$$\begin{aligned} [U, S, V] &= \text{svd}(A); \\ s &= \text{diag}(S); s1 = [1 ./ s(1:7); \text{zeros}(1, 1)]; \\ S1 &= [\text{diag}(s1); \text{zeros}(8, 8)]; \\ fh &= V * S1 * U^t * p; \text{reshape}(fh, 4, 4) \end{aligned}$$

Dans cet exemple $K = 7$ et la solution IG peut être calculée par :

$$fh = \text{svdpca}(A, p, .1, 7); \text{reshape}(fh, 4, 4)$$

$$\hat{f} = \begin{bmatrix} -0.2500 & 0.2500 & 0.2500 & -0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ -0.2500 & 0.2500 & 0.2500 & -0.2500 \end{bmatrix}$$

ou encore par l'algorithme itératif suivant:

```
for k=1:100;
fh=fh+.1*A'*(p-A*fh(:));
end;
reshape(fh,4,4)
```

$$\hat{f} = \begin{bmatrix} -0.2500 & 0.2500 & 0.2500 & -0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ -0.2500 & 0.2500 & 0.2500 & -0.2500 \end{bmatrix}$$

Notons aussi que le noyau de la transformation linéaire $g = Af$, i.e., $\{f | Af = 0\}$ est

$$V(I - S^+ S)z = \sum_{k=K+1}^N z_k v_k$$

avec z un vecteur arbitraire. Ceci nous permet de trouver toutes les solutions possibles du problème en rajoutant ce terme arbitraire à la solutions IG.

7. RÉGULARISATION

Notons que, $A^T A + \lambda I$ et $AA^T + \lambda I$ sont inversibles pour $\lambda > 0$. Ceci nous permet de calculer

```
lambda=.01;
fh=inv(AtA+lambda*eye(size(AtA)))*(A'*p);
reshape(fh,4,4)
```

$$\hat{f} = \begin{bmatrix} -0.2491 & 0.2497 & 0.2497 & -0.2491 \\ 0.2497 & 0.7484 & 0.7484 & 0.2497 \\ 0.2497 & 0.7484 & 0.7484 & 0.2497 \\ -0.2491 & 0.2497 & 0.2497 & -0.2491 \end{bmatrix}$$

ou encore

```
lambda=.01;
fh=A'*inv(AAt+lambda*eye(size(AAt)))*p;
reshape(fh,4,4)
```

qui fourni la même solution.

8. CONTRAINTE DE POSITIVITÉ

On peut remarquer que la manque d'information dans les données est telle que la contrainte de norme minimal ne restreint pas suffisamment l'espace des solutions possibles. Dans les problèmes inverses en imagerie, une information qui est souvent disponible est la positivité de la solution. Imposer alors à la solution d'être positive est alors une technique souvent utilisée. Une approche simple dans les méthodes itérative pour imposer cette contraintes est simplement l'imposer à chaque itération :

```
for k=1:100
fh=fh+.1*A'*(p-A*fh(:));
fh=fh.*(fh>0);
end
reshape(fh,4,4);
```

$$fh = \begin{bmatrix} 0 & 0.0000 & 0.0000 & 0 \\ 0.0000 & 1.0000 & 1.0000 & 0.0000 \\ 0.0000 & 1.0000 & 1.0000 & 0.0000 \\ 0 & 0.0000 & 0.0000 & 0 \end{bmatrix}$$

Bien entendu, ceci n'est qu'une méthode simple et il existe un grand nombre d'algorithmes d'optimisation sous contraintes que l'on peut utiliser, mais la description de ces algorithmes sort du cadre de ce travail.

9. MISE EN OEUVRE DANS UN CAS RÉEL

Examinons maintenant le cas d'une image de plus grande taille (256×256). Ici, il n'est pas question de former la matrice A car de dimensions ($256^2 \times 256$). Par contre, on aura besoin de calculer Af et $A^T g$, mais pour cela on n'a pas besoins de construire réellement la matrice A . Les deux fonctions suivantes effectuent ces deux tâches :

```
function p=direct(f);
p1=sum(f);
p2=sum(f');
p=[p1(:);p2(:)];
return

function f=transp(p);
l=length(p);p1=p(1:l/2);p2=p(l/2+1:l);
f=ones(l/2,1)*p1'+p2*ones(1,l/2);
return
```

Ces routines peuvent alors facilement utilisées pour obtenir les images de la figure qui suit.

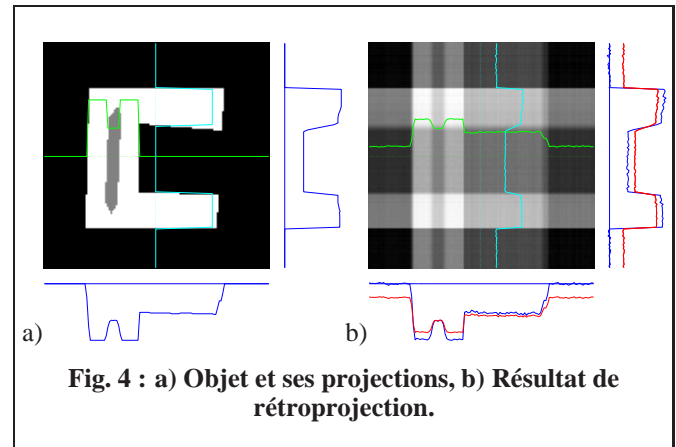


Fig. 4 : a) Objet et ses projections, b) Résultat de rétroprojection.

Nous pouvons aussi utiliser ces routines pour mettre en oeuvre la plupart des méthodes itératives :

Moindre carré avec contrainte de positivité :

```
alpha=.1;
for k=1:100
g=trans(p-direct(fh));
fh=fh+alpha*g;
fh=fh.*(fh>0);
end
```

Régularisation quadratique avec contrainte de positivité :

```
alpha=.1;d=[-1 0 -1;0 4 0;-1 0 -1];
for k=1:100
```

```

g0=trans(p-direct(fh);
g=g0-lambda*conv2(fh,d,'same');
fh=fh+alpha*g
fh=fh.*(fh>0);
end

```

Remarquons que les algorithmes présentés plus haut sont assez rudimentaires (gradient à pas constant et à nombre d'itérations fini). Il est évident que l'on peut faire mieux. À titre d'exemple, nous avons développé un logiciel d'optimisation (gpave) un peu plus élaboré qui met en œuvre d'autres algorithmes d'optimisation, comme par exemple, gradient à pas adaptative, gradient conjugué et d'autres encore.

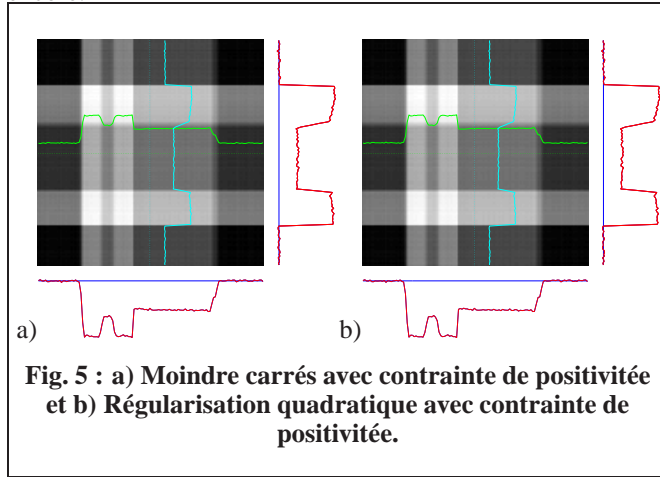


Fig. 5 : a) Moindre carrés avec contrainte de positivité et b) Régularisation quadratique avec contrainte de positivité.

Les lignes de codes Matlab qui suivent montrent l'usage de ce logiciel. Il faut tout d'abord écrire deux routines qui calculent le critère crit qui calcule

$$J = \|g - Af\|^2 + \lambda \|Df\|^2$$

et son gradient dcrit

$$\nabla J = -2A'(g - Af) + 2\lambda D'Df$$

où Df correspond à l'application d'une opération de convolution de l'image $f(i, j)$ avec une réponse impulsionnelle

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

ce qui correspond à

$$\sum_i \sum_j (|f(i, j) - f(i-1, j)|^2 + |f(i, j) - f(i, j-1)|^2).$$

Notez aussi que $D'Df$ correspond à l'application d'une opération de convolution de l'image $f(i, j)$ avec une réponse impulsionnelle

$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & -4 & 0 \\ -1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} * \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

où * signifie une convolution.

```

function J=crit(fh,p,lambda)
dp=p-direct(fh);
J0=sum(dp(:).^2);

```

```

d=[-1 1;1 -1];
df=conv2(fh,d,'same');
J1=sum(df(:).^2);
J=J0+lambda*J1;
return

```

```

function dJ=dcrit(fh,p,lambda)
dp=p-direct(fh);
dJ0=-2*trans(dp);
d=[-1 0 -1;0 4 0;-1 0 -1];
dJ1=conv2(fh,d,'same');
dJ=dJ0+lambda*dJ1;
return

```

Avec ces deux routines, le programme de la reconstruction devient très simple:

```

f0=trans(p);
options = goptions;
lambda=1;
fh=gpav('crit',f0,options,'dcrit',p,lambda);
reshape(fh,4,4)

```

Avec ce programme d'optimisation il est alors facile de modifier les routines crit et dcrit pour changer le critère de la régularisation. Les figures suivantes montrent un certain nombre des résultats.

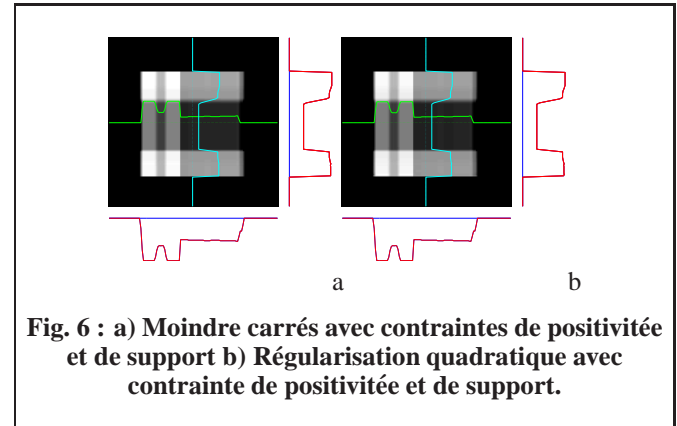
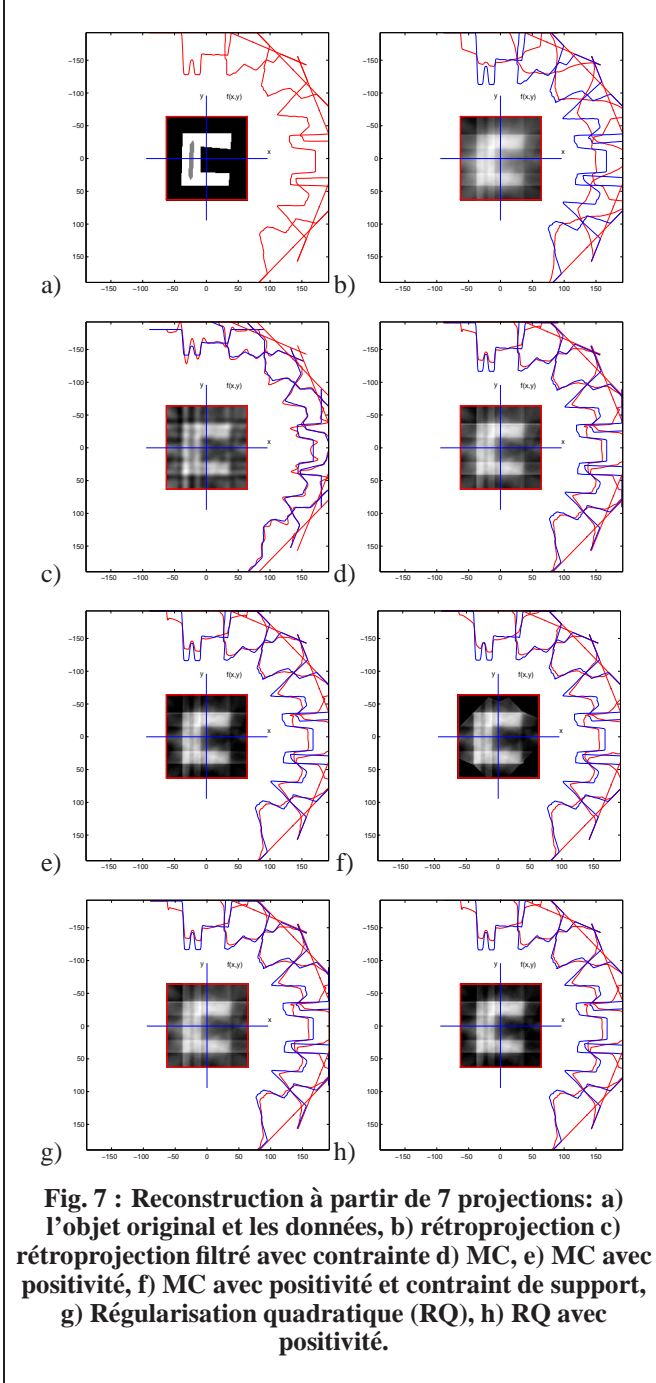


Fig. 6 : a) Moindre carrés avec contraintes de positivité et de support b) Régularisation quadratique avec contrainte de positivité et de support.

On remarque que le problème est très mal-conditionné au sens que la manque d'information dans les données est trop important. Il faut pouvoir obtenir d'autres données, i.e., des projections suivant d'autres angles. Pour cela il faut réécrire les routines directe et transp et les routines correspondantes crit et dcrit afin de pouvoir implémenter le cas général du calcul des projections suivant n'importe quel angle et la rétroprojection associée. Ces routines sont bien sûr plus complexes car elles ont besoin de toutes les paramètres géométriques; les positions et les nombre des sources et des capteurs, les positions et les pas de discrétisation suivant les axes de l'objets, etc.

Nous avons développé un ensemble de routines Matlab, sous la forme d'un toolbox [?], qui permet de simuler les situations de géométrie parallèle ou conique pour des opérateurs de projections et rétroprojection, ainsi que la mise en œuvre des méthodes de reconstructions déjà mentionnées dans ce contexte général. Les figures suivantes montrent des exemples de reconstructions pour le cas où on a 7 projections.



10. MODÉLISATION PAR CHAMPS DE MARKOV COMPOSITES

Évidemment, plus on a des données bien réparties, mieux sera les résultats. Mais, lorsque l'obtention d'autres projections est impossible, il faudra récompenser la manque d'information par des modélisations plus précises. En particulier, dans le domaine du contrôle non destructif (CND), une information *a priori* importante est que l'objet est composé d'un nombre fini de matériaux. Ceci signifie que l'image que nous cherchons à reconstruire est composée d'un nombre fini de régions homogènes. C'est exactement la modélisation de cette information *a priori* qui est l'originalité des travaux que nous menons dans notre laboratoire.

L'outil est la modélisation probabiliste par champs de Markov et l'estimation bayésienne. Un grand nombre de travaux ont été fait sur ce sujet (voir par exemple [10, 11, 12]). Ici, nous mentionnons seulement deux modélisations : Modélisation de l'image par un champs composite (intensités-contours) ou (intensités-régions). Dans la première, on introduit une variable cachée binaire $q(\mathbf{r})$ qui représente les contours et dans la deuxième on introduit une variable cachée discrète $z(\mathbf{r})$ qui peut prendre des valeurs discrètes $k = 1, \dots, K$, représentant les labels attribués aux pixels $f(\mathbf{r})$ de l'image ayant les mêmes propriétés (par exemple se trouvant dans une même région homogène).

10.1 Modèle Intensités-Contours

Dans cette modélisation, l'idée de base est de modéliser le fait qu'une image est en faite une fonction $f(\mathbf{r})$ qui est continue par morceaux (piecewise continuous) ou par régions. Il y a donc des discontinuités (contours). On peut alors modéliser ces contours par une image binaire $q(\mathbf{r})$. Le point essentiel est alors de décrire à l'aide d'une loi de probabilité conditionnelle $p(\mathbf{f}|\mathbf{q})$, le lien qu'il y entre des variables intensités \mathbf{f} et des variables contours \mathbf{q} qui peut être résumé par :

Cas 1D:

$$p(f_j|q_j, f_i, i \neq j) = \mathcal{N}(f_j|\beta(1-q_j)f_{j-1}, \sigma_f^2)$$

où nous avons utilisé la notation $\mathcal{N}(x|m, v)$ pour représenter une distribution gaussienne de la variable x avec la moyenne m et la variance v .

Cas 2D:

$$p(f(\mathbf{r})|q(\mathbf{r}), f(\mathbf{s})) = \mathcal{N}\left(f(\mathbf{r})|\beta(1-q(\mathbf{r})) \sum_{\mathbf{s} \in \mathcal{V}(\mathbf{r})} f(\mathbf{s}), \sigma_f^2\right)$$

Ensuite, en choisissant une loi *a priori* appropriée pour $p(\mathbf{q})$ et en utilisant des lois $p(\mathbf{g}|\mathbf{f})$ et $p(\mathbf{f}|\mathbf{q})$, on obtien la loi *a posteriori* $p(\mathbf{f}, \mathbf{q}|\mathbf{g})$ qui peut être utilisée pour inferer conjointement sur \mathbf{f} et sur \mathbf{q} . A titre d'indication, considérons l'estimation au sens du MAP :

$$(\hat{\mathbf{f}}, \hat{\mathbf{q}}) = \arg \max_{\mathbf{f}, \mathbf{q}} \{p(\mathbf{f}, \mathbf{q}|\mathbf{g})\}$$

qui peut être obtenu par un algorithme itératif du type :

$$\begin{aligned} \hat{\mathbf{f}} &= \arg \max_{\mathbf{f}} \{p(\mathbf{f}|\mathbf{g}, \mathbf{q})\} = \arg \min_{\mathbf{f}} \{J(\mathbf{f})\} \\ \hat{\mathbf{q}} &= \arg \max_{\mathbf{q}} \{p(\mathbf{q}|\mathbf{g})\} \end{aligned}$$

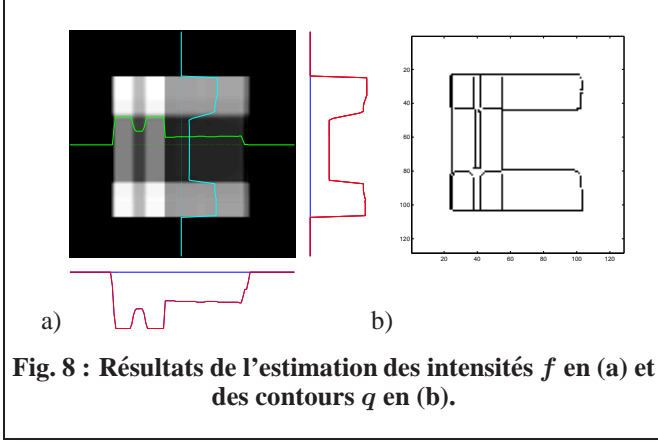
avec

$$J(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \sum_{\mathbf{r}} (1-q(\mathbf{r})) \left(f(\mathbf{r}) - \beta \sum_{\mathbf{s} \in \mathcal{V}(\mathbf{r})} f(\mathbf{s}) \right)^2$$

L'étape difficile est l'obtention de l'expression de $p(\mathbf{q}|\mathbf{g})$ et surtout son optimisation, qui idéalement ne peut se faire qu'à l'aide d'une recherche combinatoire. Il existe un très grand nombres de travaux portant sur différentes approximations qui permettent d'effectuer cette optimisation d'une manière approchée mais réaliste en coût de calcul pour des applications réelles.

Pour plus de détail sur cette méthode se référer à [10, 11, 12].

Ici, nous montrons un résultat typique que l'on peut obtenir avec de telles méthode. Comme nous pouvons constater, cette modélisation *a priori* n'est pas encore suffisamment forte pour obtenir un résultat satisfaisant pour ce problème inverse très difficile.



10.2 Modèle Intensités-Régions

La modélisation précédente, bien que déjà plus spécifique, n'apportait pas d'information sur des valeurs qui peuvent prendre des pixels de l'image. Dans certaines applications, par exemple en contrôle non destructif (CND), nous savons *a priori* que l'objet est composé d'un nombre fini de matériaux (air, métal, composite). La modélisation qui suit permet de prendre en compte ce type d'information *a priori*.

Plus spécifiquement, nous proposons de modéliser l'image par un champs composite (intensités-labels), où les labels $z(\mathbf{r})$ représentent la nature de matériaux à la position du pixel \mathbf{r} et l'homogénéité des intensités $\mathbf{f}_k = \{f(\mathbf{r}), \mathbf{r} \in \mathcal{R}_k\}$ dans une région données $\mathcal{R}_k = \{\mathbf{r} : z(\mathbf{r}) = k\}$ est modélisé par un champs de Gauss-Markov:

$$p(\mathbf{f}_k) = \mathcal{N}(\mathbf{f} | m_k \mathbf{1}, \Sigma_k)$$

ce qui peut être interprété aussi par des relations suivantes:

$$\begin{aligned} p(f(\mathbf{r}) | z(\mathbf{r}) = k) &= \mathcal{N}(f(\mathbf{r}) | m_k, v_k) \\ p(\mathbf{f}(\mathbf{r})) &= \sum_{k=1}^K p(z(\mathbf{r}) = k) \mathcal{N}(f(\mathbf{r}) | m_k, \sigma_k^2) \\ \Sigma_k &= \text{diag}[v_1, \dots, v_K] \end{aligned}$$

ce qui montre que la distribution marginale des pixels de l'images est modélisée par un mélange de gaussiennes.

Supposant ensuite qu'*a priori* les pixels qui se trouvent dans deux régions différentes soient indépendantes. On peut alors écrire :

$$\begin{aligned} p(\mathbf{f} | \mathbf{z}) &= \prod_{k=1}^K \mathcal{N}(\mathbf{f} | m_k \mathbf{1}, \Sigma_k) \\ &= \prod_{k=1}^K \prod_{\mathbf{r} \in \mathcal{R}_k} \mathcal{N}(f(\mathbf{r}) | m_k, v_k) \end{aligned}$$

La particularité de la méthode que nous proposons est une modélisation spécifique pour des labels des régions (Champs de Potts) qui permet de décrire pour $p(\mathbf{z})$

$$p(\mathbf{z}) \propto \exp \left[\alpha \sum_{\mathbf{r} \in \mathcal{R}} \sum_{\mathbf{s} \in \mathcal{V}(\mathbf{r})} \delta(z(\mathbf{r}) - z(\mathbf{s})) \right]$$

Notons par $\theta = \{\sigma_\varepsilon^2, (m_k, \sigma_k^2), k = 1, \dots, K\}$, appelé le vecteur des hyperparamètres. Dans un problème réel (reconstruction non supervisé), ce vecteur est aussi inconnu et il faut l'estimer. Il faut alors lui attribuer une loi *a priori* $p(\theta)$. Une fois fait, nous avons tous les éléments en main pour exprimer la loi *a posteriori* jointe

$$\begin{aligned} p(\mathbf{f}, \mathbf{z}, \theta | \mathbf{g}) &\propto p(\mathbf{f} | \mathbf{z}, \theta, \mathbf{g}) p(\mathbf{z}) p(\theta) \\ &\propto p(\mathbf{g} | \mathbf{f}, \theta) p(\mathbf{f} | \mathbf{z}, \theta) p(\mathbf{z}) p(\theta). \end{aligned}$$

On peut ensuite utiliser cette loi pour inférer toutes ces inconnues. Cette étape se fait en général par l'intermédiaire de la définition des estimateurs $\hat{\mathbf{f}}$, $\hat{\mathbf{z}}$ et $\hat{\theta}$ qui correspondent soit au mode ou la moyenne de cette lois *a posteriori*, qui ne peuvent, en général être calculés que par des algorithmes itératives qui utilisent successivement les lois *a posteriori* conditionnelles

$$\begin{aligned} p(\mathbf{f} | \mathbf{z}, \theta, \mathbf{g}) &\propto p(\mathbf{g} | \mathbf{f}, \theta) p(\mathbf{f} | \mathbf{z}, \theta) \\ p(\mathbf{z} | \mathbf{f}, \theta, \mathbf{g}) &\propto p(\mathbf{z} | \mathbf{f}, \theta) p(\mathbf{z}) \\ p(\theta | \mathbf{f}, \mathbf{z}, \mathbf{g}) &\propto p(\theta | \mathbf{z}, \theta, \mathbf{g}) p(\theta) \end{aligned}$$

Différents choix sont alors possibles. Ici, nous mentionnons ceux que nous avons implémentés et utilisés:

- MAP (Algorithm 1):

$$\begin{cases} \hat{\mathbf{f}} = \arg \max_{\mathbf{f}} \{p(\mathbf{f} | \mathbf{z}, \theta, \mathbf{g})\} \\ \hat{\mathbf{z}} = \arg \max_{\mathbf{z}} \{p(\mathbf{z} | \mathbf{f}, \theta, \mathbf{g})\} \\ \hat{\theta} = \arg \max_{\theta} \{p(\theta | \mathbf{f}, \mathbf{z}, \mathbf{g})\} \end{cases}$$

- Gibbs (Algorithm 2):

$$\begin{cases} \text{échant. } \hat{\mathbf{f}} & \text{avec } p(\mathbf{f} | \mathbf{z}, \theta, \mathbf{g}) \\ \text{échant. } \hat{\mathbf{z}} & \text{avec } p(\mathbf{z} | \mathbf{f}, \theta, \mathbf{g}) \\ \text{échant. } \hat{\theta} & \text{avec } p(\theta | \mathbf{f}, \mathbf{z}, \mathbf{g}) \end{cases}$$

- MAP-Gibbs (Algorithm 3):

$$\begin{cases} \hat{\mathbf{f}} = \arg \max_{\mathbf{f}} \{p(\mathbf{f} | \mathbf{z}, \theta, \mathbf{g})\} \\ \text{échant. } \hat{\mathbf{z}} & \text{avec } p(\mathbf{z} | \mathbf{f}, \theta, \mathbf{g}) \\ \text{échant. } \hat{\theta} & \text{avec } p(\theta | \mathbf{f}, \mathbf{z}, \mathbf{g}) \end{cases}$$

- Gibbs-EM (Algorithm 4):

$$\begin{cases} \hat{\mathbf{f}} = \mathbb{E} \{ \mathbf{f} | \mathbf{z}, \theta, \mathbf{g} \} = \arg \max_{\mathbf{f}} \{p(\mathbf{f} | \mathbf{z}, \theta, \mathbf{g})\} \\ \hat{\mathbf{z}} = \mathbb{E} \{ \mathbf{z} | \mathbf{f}, \theta, \mathbf{g} \} \text{ ou échant. avec } p(\mathbf{z} | \mathbf{f}, \theta, \mathbf{g}) \\ \hat{\theta} = \mathbb{E} \{ \theta | \mathbf{f}, \mathbf{z}, \mathbf{g} \} \text{ ou échant. avec } p(\theta | \mathbf{f}, \mathbf{z}, \mathbf{g}) \end{cases}$$

L'équivalence de la première ligne est due au fait que la lois *a posteriori* $p(\mathbf{f} | \mathbf{z}, \theta, \mathbf{g})$ est une gaussienne et donc sa mode et sa moyenne se confondent.

Pour plus de détails sur les expressions des lois conditionnelles qui interviennent dans ces algorithmes et la mise en oeuvre de la méthode dans un cadre plus général se référer à [13, 14, 15, 16].

Principal avantage d'une telle modélisation et d'un tel méthode est que l'on obtient non seulement une estimation de \mathbf{f} mais aussi de \mathbf{z} qui représente une segmentation de l'image, et aussi par un simple algorithme de détection de contours sur \mathbf{z} on obtiendra aussi une image des contours \mathbf{q} . La figure qui suit montre un résultat typique.

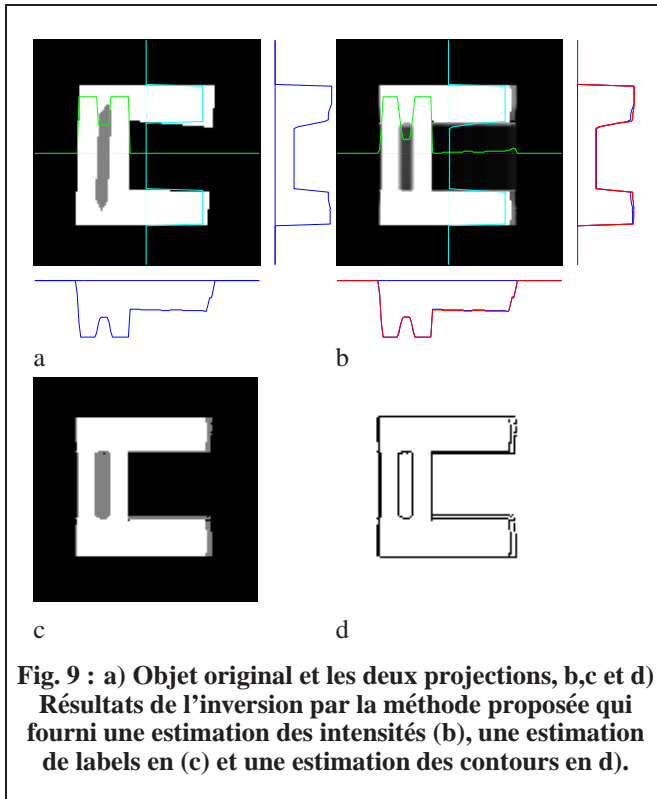


Fig. 9 : a) Objet original et les deux projections, b,c et d) Résultats de l'inversion par la méthode proposée qui fourni une estimation des intensités (b), une estimation de labels en (c) et une estimation des contours en d).

11. CONCLUSION

Dans ce travail, à but pédagogique, au travers d'un problème inverse de la reconstruction d'image en tomographie X lorsque le nombre de projections sont très limité, nous avons analysé les difficultés inhérentes des problèmes inverses. Le principal objectif était de montrer que les différentes méthodes classiques naïves, mais très utilisées, ne donnent pas de solutions satisfaisantes et qu'il y a un besoin de proposer des méthodes d'inversion plus sophistiquées qui permettent d'introduire de l'information *a priori* nécessaire pour compenser la manque d'information dans les données.

Un grand nombre de modélisations ont été proposées (voir par exemple [17, 18]). Mais, ici, nous nous sommes contenté des méthodes qui modélisent l'image au niveau des pixels.

Le cas particulier de la reconstruction à partir de deux projections est détaillé et une méthode basée sur la modélisation de l'image par un champs de Markov composé (intensité-labels) et l'estimation bayésienne est présentée qui permet, au moins, d'obtenir une solution satisfaisante au problème. Les routines Matlab correspondant est disponible sur <http://djafari.free.fr/TomoX>.

REFERENCES

- [1] G. T. Herman and S. W. Rowland, "Three methods for reconstructing objects from X-rays: A comparative study," *Computer Graphics and Image Processing*, vol. 2, pp. 151–178, 1973.
- [2] K. M. Hanson, "On the optimality of the filtered backprojection algorithm," *J. Comput. Assist. Tomogr.*, vol. 4, pp. 361–363, 1980.
- [3] G. T. Herman, *Image reconstruction from projections. The fundamentals of computerized tomography*. New York, NY: Academic Press, 1980.

- [4] P. Grangeat, "Mathematical framework of cone beam 3D reconstruction via the first derivative of the Radon transform," in *Mathematical Methods in Tomography* (G. T. Herman, A. K. Louis, and F. Natterer, eds.), vol. 1497, (New York), pp. 66–97, Springer Verlag, 1991.
- [5] F. Natterer, "Algorithms in tomography," in *The State of the Art in Numerical Analysis*, Clarendon Press, duff, i.s. and watson, g.a. ed., 1997.
- [6] A. Mohammad-Djafari and C. Soussen, "Compact object reconstruction," in *Discrete Tomography: Foundations, Algorithms and Applications* (G. T. Herman and A. Kuba, eds.), ch. 14, pp. 317–342, Boston, MA: Birkhauser, 1999.
- [7] F. Natterer, "Numerical methods in tomography," in *Acta Numerica*, vol. 8, Cambridge University Press, July 1999.
- [8] G. Demoment, J. Idier, J.-F. Giovannelli, and A. Mohammad-Djafari, "Problèmes inverses en traitement du signal et de l'image," vol. TE 5 235 of *Traité Télécoms*, pp. 1–25, Paris, France: Techniques de l'Ingénieur, 2001.
- [9] G. Demoment and J. Idier, "Régularisation d'un problème mal posé : approches gnriques," in *Approche bayésienne pour les problèmes inverses* (J. Idier, ed.), (Paris), pp. 41–58, Trait IC2, Srie traitement du signal et de l'image, Herms, 2001.
- [10] M. Nikolova, A. Mohammad-Djafari, and J. Idier, "Inversion of large-support ill-conditioned linear operators using a Markov model with a line process," in *Proc. IEEE ICASSP*, vol. V, (Adelaide, Australia), pp. 357–360, 1994.
- [11] S. S. Saquib, C. A. Bouman, and K. D. Sauer, "ML parameter estimation for Markov random fields with applications to Bayesian tomography," *IEEE Trans. Image Processing*, vol. 7, pp. 1029–1044, July 1998.
- [12] A. Mohammad-Djafari, "Hierarchical markov modeling for fusion of x ray radiographic data and anatomical data in computed tomography," in *Int. Symposium on Biomedical Imaging (ISBI 2002)*, 7-10 Jul., Washington DC, USA, July 2002.
- [13] H. Snoussi and A. Mohammad-Djafari, "Fast joint separation and segmentation of mixed images," *Journal of Electronic Imaging*, vol. 13, pp. 349–361, Apr. 2004.
- [14] Hichem Snoussi AND Ali Mohammad-Djafari, "Bayesian unsupervised learning for source separation with mixture of gaussians prior," *Journal of VLSI Signal Processing Systems*, vol. 37, pp. 263–279, June/July 2004.
- [15] O. Féron and A. Mohammad-Djafari, "Image fusion and joint segmentation using an MCMC algorithm," *Journal of Electronic Imaging*, vol. 14, p. paper no. 023014, Apr 2005.
- [16] F. Humblot and A. Mohammad-Djafari, "Super-Resolution using Hidden Markov Model and Bayesian Detection Estimation Framework," *EURASIP Journal on Applied Signal Processing*, vol. Special number on Super-Resolution Imaging: Analysis, Algorithms, and Applications, pp. ID 36971, 16 pages, 2006.
- [17] A. Kuba and G. T. Herman, "Discrete tomography: A historical overview," in *Discrete Tomography: Foundations, Algorithms and Applications* (G. T. Herman and A. Kuba, eds.), ch. 1, pp. 3–34, Boston, MA: Birkhauser, 1999.
- [18] Charles Soussen and Ali Mohammad-Djafari, "Polygonal and polyhedral contour reconstruction in computed tomography," *IEEE Trans. on Image Processing*, vol. 13, pp. 1507–1523, Nov 2004.